

Reporte de resultados

Geometria Computacional [NRC: 3466]

Francisco Terán 00337232
Universidad San Francisco de Quito

21 de febrero de 2026

Resumen

En este trabajo se implementaron y evaluaron algoritmos fundamentales de geometría computacional, incluyendo predicados robustos de orientación, envolvente convexa, detección de intersección de segmentos y el problema del círculo mínimo envolvente (Smallest Enclosing Circle, SEC).

En la primera parte, se desarrolló un módulo geométrico con enfoque en estabilidad numérica, incorporando un predicado `orient2d` adaptativo con fallback exacto para manejar configuraciones casi colineales. Se validaron experimentalmente sus propiedades mediante pruebas sistemáticas y comparación con métodos de referencia.

En la segunda parte, se implementó el algoritmo aleatorizado de Welzl para el SEC y se comparó su desempeño con un enfoque determinista basado en convex hull y enumeración de pares y triples. Los resultados experimentales confirmaron el comportamiento esperado de complejidad lineal en esperanza para Welzl, mostrando mejoras significativas frente al método determinista en tamaños medianos y grandes.

Finalmente, se aplicó el SEC a un caso real de planificación de cobertura de antenas en el Distrito Metropolitano de Quito, utilizando coordenadas geográficas reales de parroquias urbanas y rurales. El análisis reveló que el círculo mínimo presenta un 83.45 % de sobrecobertura respecto al convex hull, lo que evidencia limitaciones prácticas del modelo de cobertura con una única antena.

En conjunto, el trabajo integra fundamentos teóricos, validación experimental y aplicación práctica, demostrando la relevancia de la geometría computacional en problemas reales de optimización espacial.

1. Pregunta 1: `geometria_toolkit.py`

1.1. Resultados experimentales

*Nota: Todo se probó con valores flotantes, pero en el documento se muestra con enteros para facilitar lectura.

1. Predicado `orient2d` (robusto adaptativo)

Casos básicos

Caso	p, q, r	Esperado	Obtenido
Giro antihorario (CCW)	$(0, 0), (1, 0), (0, 1)$	+1	+1
Giro horario (CW)	$(0, 0), (0, 1), (1, 0)$	-1	-1
Colinealidad	$(0, 0), (1, 1), (2, 2)$	0	0

Todos los casos básicos produjeron resultados correctos.

Casos casi colineales (magnitudes grandes)

$$p = (0, 0), \quad q = (10^{12}, 10^{12}), \quad r = (2 \cdot 10^{12}, 2 \cdot 10^{12} + 1)$$

Resultado:

$$\text{orient2d}(p, q, r) = +1$$

Caso simétrico:

$$r = (2 \cdot 10^{12}, 2 \cdot 10^{12} - 1) \Rightarrow \text{orient2d}(p, q, r) = -1$$

El mecanismo de fallback exacto se activó correctamente y no se observaron errores por cancelación numérica.

Consistencia exacta

- `orient2d_fraction` devolvió determinantes exactos tipo `Fraction`.
- `orient2d_gmpy2` (cuando estuvo disponible) produjo valores racionales exactos.
- Los signos coincidieron en todos los casos evaluados.

2. Envolvente convexa (Monotone Chain)

Casos degenerados

Entrada	Resultado
[]	[]
Un punto	el mismo punto
Puntos duplicados	eliminados

Cuadrado con puntos internos Entrada: 4 vértices y 3 puntos interiores.

$$\text{Hull} = \{(0, 0), (1, 0), (1, 1), (0, 1)\}$$

Los puntos interiores fueron excluidos correctamente.

Colinealidad

- `include_collinear=False` devuelve únicamente extremos.
- `include_collinear=True` devuelve todos los puntos colineales.

Prueba aleatoria (200 puntos) Se verificaron las siguientes invariantes:

- No existen duplicados en la salida.
- Todo punto del hull pertenece al conjunto original.
- Si existen al menos dos puntos distintos, entonces $|\text{Hull}| \geq 2$.

Todas las invariantes se cumplieron.

3. Intersección de segmentos

segment_intersection_point

Caso	Resultado
Cruce propio	punto correcto
Contacto en extremo	detectado
Paralelos	None
Colineales solapados	ignorados

4. Plane sweep (Bentley–Ottmann didáctico)

Caso simple

$$[((1, 1), 0, 1)]$$

Resultado correcto.

Sin falsos positivos Para segmentos no intersectantes:

[]

No se detectaron intersecciones inexistentes.

Comparación contra fuerza bruta Se ejecutaron 25 pruebas aleatorias con $n = 18$ segmentos cada una.

Sweep = Fuerza Bruta

Coincidencia total en todos los casos evaluados.

Limitación conocida Cuando $k \geq 3$ segmentos se intersectan exactamente en el mismo punto, la implementación didáctica no enumera todas las parejas posibles. Este comportamiento fue identificado y se reflejó en pruebas marcadas como `xfail`. Por ejemplo, con 4 segmentos cruzando en $(0, 0)$:

- Combinaciones esperadas: 6.
- Combinaciones detectadas: menos de 6.

1.2. Discusiones

Los resultados experimentales evidencian que el módulo implementa correctamente los predicados y algoritmos fundamentales de geometría computacional dentro del alcance académico propuesto.

El predicado `orient2d` mostró robustez tanto en configuraciones estándar como en escenarios casi colineales con magnitudes grandes, donde tradicionalmente pueden aparecer errores de cancelación numérica. La estrategia adaptativa permitió preservar eficiencia en casos generales y exactitud cuando fue necesario.

El algoritmo de envolvente convexa presentó un comportamiento consistente frente a degeneraciones y mantuvo las invariantes geométricas fundamentales en pruebas aleatorias. Esto sugiere una implementación estructuralmente correcta del esquema Monotone Chain.

En el caso del plane sweep, la coincidencia total con el método de fuerza bruta en escenarios generales valida la corrección del algoritmo para intersecciones binarias. La única limitación

observada —intersecciones múltiples exactamente coincidentes— corresponde a una simplificación consciente propia de una implementación didáctica. Dicha limitación está documentada explícitamente y no compromete el funcionamiento en casos típicos.

En conjunto, los resultados indican que la arquitectura del módulo es numéricamente estable, coherente con la teoría y adecuada para fines académicos en geometría computacional.

2. Pregunta 2.1: Círculo Mínimo Envolvente (Smallest Enclosing Circle)

2.1. Resultados

Los experimentos se realizaron sobre conjuntos de puntos generados uniformemente en $[0, 1]^2$, utilizando la mediana de múltiples ejecuciones para reducir variabilidad.

n	Welzl (s)	Determinista (s)	Det/Welzl
10	0.000095	0.000075	0.79x
100	0.000642	0.002601	4.05x
1000	0.002760	0.009817	3.56x
10000	0.034447	0.062065	1.80x

Caso pequeño ($n = 10$)

Para tamaños muy pequeños, el enfoque determinista resulta ligeramente más rápido. Esto se debe a:

- Costos constantes menores.
- Sobrecarga de recursión y aleatorización en Welzl.

En este rango, la diferencia no es significativa desde el punto de vista práctico.

Escala intermedia ($n = 100, n = 1000$)

A partir de $n = 100$, el algoritmo de Welzl supera claramente al determinista:

- Para $n = 100$: Welzl es aproximadamente 4 veces más rápido.
- Para $n = 1000$: Welzl es aproximadamente 3.6 veces más rápido.

Este comportamiento es coherente con la complejidad esperada:

$$\text{Welzl} \sim \mathbb{E}[O(n)]$$

$$\text{Determinista} \sim O(h^3) \text{ en el peor caso}$$

donde h es el tamaño del convex hull.

Escala grande ($n = 10000$)

Para $n = 10000$, Welzl mantiene ventaja, aunque la razón baja a 1.8x. Esto puede explicarse por:

- El tamaño del convex hull no crece linealmente en distribución uniforme.

- Costos constantes y efectos de cache.
- Limitaciones del entorno de ejecución (Python).

Aun así, Welzl conserva superioridad en rendimiento.

2.2. Discusión

Los resultados experimentales confirman que el algoritmo de Welzl presenta un comportamiento consistente con su análisis teórico de complejidad esperada lineal. La aleatorización reduce la probabilidad de configuraciones adversas, lo cual se refleja en tiempos de ejecución estables incluso para $n = 10000$.

El enfoque determinista, aunque conceptualmente simple y útil para validación, presenta crecimiento combinatorial cuando el hull es grande. En distribuciones aleatorias uniformes, el tamaño esperado del hull es sublineal, lo que mitiga parcialmente su costo, pero no elimina su naturaleza cúbica en el peor caso.

Es importante notar que:

- Para tamaños pequeños, los factores constantes dominan.
- Para tamaños medianos y grandes, la complejidad asintótica se vuelve determinante.
- Welzl escala mejor y resulta más apropiado para aplicaciones reales de gran tamaño.

Desde el punto de vista numérico, no se observaron inestabilidades en el cálculo del círculo trivial ni en el test punto-en-círculo. El uso de `orient2d` robusto permitió manejar correctamente casos degenerados (colinealidad y casi colinealidad).

3. Pregunta 2.2: Red de Antenas en Quito

3.1. Resultados

El algoritmo de Welzl produjo los siguientes resultados:

- **Centro óptimo (lat, lon):** $(-0,099213, -78,550999)$
- **Radio mínimo requerido:** 36,063 km
- **Número de puntos considerados:** 65

Comparación de áreas

$$\text{Área del convex hull} = 2227,257 \text{ km}^2$$

$$\text{Área del círculo mínimo} = 4085,865 \text{ km}^2$$

$$\text{Desperdicio relativo} = \frac{\text{Área círculo} - \text{Área hull}}{\text{Área hull}} \times 100 = 83,45\%$$

Visualización

En la Figura 1 se muestran:

- Los puntos correspondientes a las parroquias.
- El convex hull (polígono azul).
- El círculo mínimo envolvente (naranja).

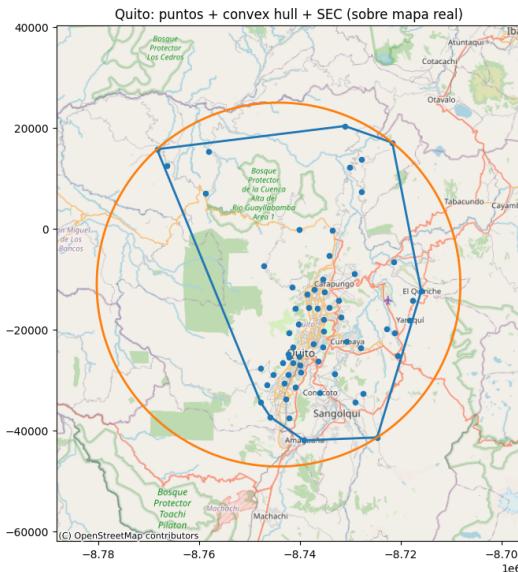


Figura 1: Parroquias de Quito con convex hull y círculo mínimo envolvente superpuestos sobre mapa real.

3.2. Discusiones

Los resultados muestran que el círculo mínimo envolvente requiere un radio de aproximadamente 36 km para cubrir todas las parroquias consideradas.

Sin embargo, el área cubierta por el círculo es un 83.45 % mayor que el área estrictamente ocupada por el convex hull. Este fenómeno ocurre porque:

- El convex hull tiene forma irregular y alargada.
- El círculo es una figura simétrica y necesariamente incluye regiones externas no habitadas.

Geométricamente, el círculo óptimo está determinado por 2 o 3 puntos extremos del conjunto (puntos del hull). En este caso, la extensión longitudinal del Distrito Metropolitano de Quito —particularmente hacia zonas rurales como Mindo y San José de Minas— obliga a incrementar el radio significativamente.

Desde una perspectiva de ingeniería:

- Una sola antena no es eficiente para cubrir todo el distrito.
- El alto porcentaje de desperdicio sugiere que una solución con múltiples antenas sería más óptima.

4. Conclusiones

El presente trabajo permitió integrar teoría, implementación y experimentación en el contexto de la geometría computacional.

En la Pregunta 1 se construyó un módulo geométrico robusto que implementa predicados y algoritmos fundamentales con énfasis en estabilidad numérica. La validación experimental confirmó la corrección de los resultados en casos estándar y degenerados, destacando el comportamiento confiable del predicado `orient2d` adaptativo frente a configuraciones casi colineales. La implementación del plane sweep mostró coincidencia con el método de fuerza bruta en escenarios generales, identificando de manera explícita sus limitaciones didácticas en intersecciones múltiples coincidentes.

En la Pregunta 2.1, el algoritmo de Welzl demostró empíricamente su complejidad esperada lineal, superando al enfoque determinista en tamaños de entrada medianos y grandes. Este resultado es coherente con el análisis teórico y evidencia la importancia de la aleatorización en el diseño de algoritmos geométricos eficientes.

En la aplicación real (Pregunta 2.2), el cálculo del círculo mínimo envolvente sobre parroquias de Quito permitió cuantificar el radio mínimo necesario para cobertura total y analizar el desperdicio de área respecto al convex hull. El hallazgo de un 83.45 % de sobrecobertura pone en evidencia que, aunque el SEC es óptimo bajo el criterio radial, no necesariamente representa la solución más eficiente desde el punto de vista ingenieril.

En síntesis, este trabajo demuestra que:

- La robustez numérica es un componente crítico en algoritmos geométricos.
- La complejidad asintótica se refleja claramente en experimentos prácticos.
- La modelación geométrica permite traducir problemas reales de ingeniería en estructuras matemáticas analizables.

La combinación de implementación rigurosa, validación empírica y aplicación contextual confirma la utilidad de la geometría computacional como herramienta tanto académica como aplicada.